Федеральное государственное бюджетное образовательное учреждение высшего образования

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

Кафедра информационных систем и программной инженерии

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту по дисциплине   
"Технологии разработки мобильных приложений"

на тему

Разработка прототипа мобильного приложения

«Многозадачный календарь»

Выполнил: студент группы ПРИ-120 Парахин К.В.

Приняла: преп. кафедры ИСПИ

Рощина А.И.

Владимир, 2023

**Аннотация**

**ABSTRACT**

**Содержание**

[ВВЕДЕНИЕ 3](#_Toc146879750)

[ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ 4](#_Toc146879751)

[Описание предметной области 4](#_Toc146879752)

[Функциональная декомпозиция системы 7](#_Toc146879753)

[Описание состава данных 10](#_Toc146879754)

[Требования к разрабатываемой системе 12](#_Toc146879755)

[РАЗРАБОТКА ПРИЛОЖЕНИЯ 14](#_Toc146879756)

[ТЕСТИРОВАНИЕ СИСТЕМЫ 15](#_Toc146879757)

[Модульное тестирование 15](#_Toc146879758)

[Репозиторий проекта 16](#_Toc146879759)

[Заключение 17](#_Toc146879760)

[Список использованных источников 18](#_Toc146879761)

[ПРИЛОЖЕНИЕ А. Схема работы алгоритма 19](#_Toc146879762)

# ВВЕДЕНИЕ

Для начала выполнения разработки прототипа информационной системы «Многозадачный календарь» требуется выполнить следующие шаги по проектированию:

- описать предметную область

- рассмотреть основные аналоги, провести их сравнение

- выделить основные термины предметной области, сформировать словарь предметной области

- выделить ее основные функциональные требования, с помощью основных UML-диаграмм описать составы данных и категории рассматриваемой системы

- провести описание основных прецедентов пользователей, сущности системы

- провести динамическое моделирование системы

- выбрать средства разработки и основные фреймворки, требующиеся для реализации функционала

- выполнить разработку в репозитории, провести тестирование полученного приложения

# ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Описание предметной области

Предметная область была названа: «Многозадачный календарь» (потому что объединяет функции как личного календаря, так и делового ToDo календаря, который интегрируется с действиями пользователей в группе).

«Почему многозадачный календарь действительно полезен для современного человека?»

Ведение своего личного, а также рабочего календаря является очень актуальной задачей для школьников, студентов, рабочих специалистов и остальных групп людей. Жизнь каждого из этих людей каждый день состоит из множества занятий, встреч, поездок и большого «груза» самостоятельной работы. Чтобы упростить жизнь современным людям, нужно дать им возможность учета своей активности и занятости, а именно: вести свои занятия, планировать мероприятия и события, а также задачи и самостоятельную работу, которые требуется выполнить в определенный срок. Особенно удобно это будет сделать в виде мобильного приложения, причем вход в свою учетную запись можно совершать не только с одного устройства, а с любого, которое окажется под рукой. А если ввести возможность планирования не только сугубо одних рабочих (или наоборот учебных) мероприятий или задач, а впринципе любых – то пользование приложением даст дополнительные плюсы пользователям. Кроме того, оно сможет подсказать о начале мероприятий, чтобы пользователи не забывали на них присутствовать, в «дружелюбной» форме напоминать о выполнении задач и поставленных целей, а также формировать отчеты о занятости отдельного пользователя и достигнутых им результатов деятельности в различных сферах активностей.

Сравнительный обзор аналогов

Для того, чтобы погрузиться в эту тему необходимо провести сравнительный анализ нескольких сервисов, предоставляющих функционал личного многозадачного календаря.

Мною были выделены 4 основных календарных сервиса: Google Calendar, Teamup, TimeBlocks, Microsoft Outlook calendar.

Краткое описание каждого из этих сервисов:

1. Google Calendar — сервис для планирования встреч, событий и дел, разработанный компанией Google. Google Календарь стал доступен в бета-версии 13 апреля 2006 года.

Очень часто используется в IT-компаниях (например, в той, которой я сейчас работаю, удобен, прост и многозадачен). Дает возможность создавать звонки, встречи, планировать некоторые задачи и получать напоминания о мероприятиях.

2) Календарь Teamup — это простое цифровое приложение для совместного использования календаря, которое упрощает и оптимизирует общение вашей команды. Это приложение для организации и планирования является простым, безопасным и масштабируемым. Это приложение для организации и планирования позволяет пользователям обмениваться планами, событиями, расписаниями и обновлениями статуса. Вы получаете полную видимость того, кто и что делает в команде, а также автономность, создавая индивидуальный доступ к календарю для отдельных лиц и групп.

3) TimeBlocks — это многоплатформенное приложение-календарь, которое позволяет создавать структурированные повестки дня для обсуждений и встреч в команде и помогает придерживаться планов и расписаний. Эта альтернатива Google - календарю имеет простой и элегантный интерфейс, который предоставляет вам подробное представление о ваших событиях.

4) Календарь Microsoft Outlook — это хорошая альтернатива календарю Google, которая предлагает достаточное количество инструментов для удовлетворения потребностей большинства людей в календаре и планировании. Это бесплатное приложение имеет простой, незагроможденный интерфейс, который намного лучше многих платных альтернатив, доступных в Магазине Windows. Используя календарь Microsoft, вы можете создавать встречи и события, организовывать собрания, просматривать групповые расписания, просматривать календари, отправлять календари кому-либо по электронной почте и т.д.

На рисунке 1 ниже я привел скриншот пользования Google – календарем (как наиболее популярного и знакомого мне сервиса):

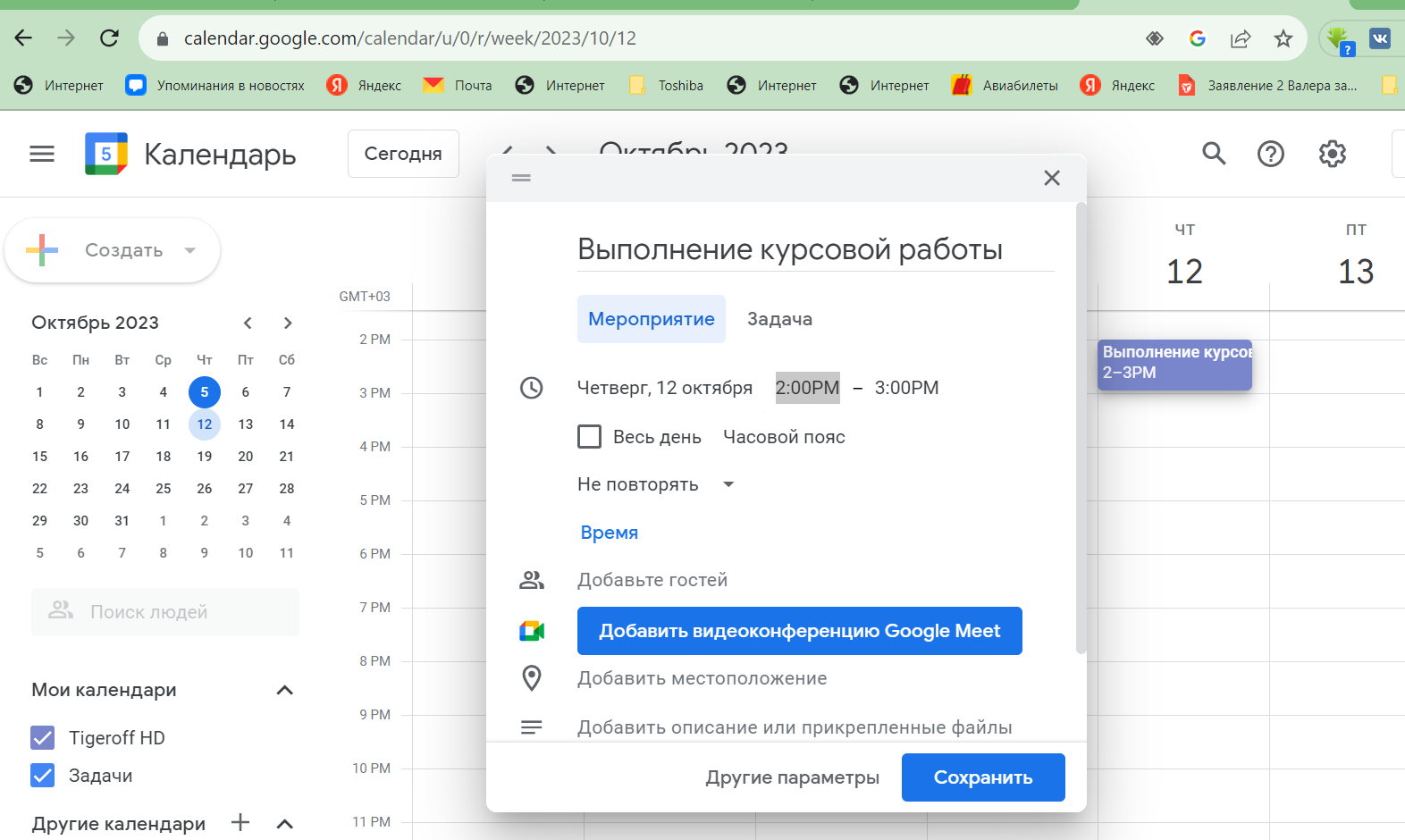


Рисунок 1. Пользование Google – календарем

Затем мною были выделены некоторые критерии, по которым я буду проводить обзор и сравнение этих сервисов:

|  |  |
| --- | --- |
| Критерии | Краткое описание |
| Дизайн | Оформление контента, совокупность всех графических элементов на веб-странице сервиса |
| Удобство использования | Понятная и доступная структура сервиса, хорошая видимость и оформление основных элементов сервиса |
| Функциональность календарного сервиса | Практичность использования календарного сервиса, наличие на нем всех нужных опций и возможностей:  - создание различных типов мероприятий  - добавление задач и целей  - возможность рассылки уведомлений и напоминаний |
| Динамичность работы | Скорость работы сервиса, то есть постоянное обновление информации на нем, адаптивность и скорость работы |
| Интеграция | Возможность интеграции сервиса планирования мероприятий в другие программные продукты. Автоматическое определение и подхватывание нового контента |

Таблица 1. Сравнительные критерии оценки сервисов

Ниже приведена таблица с сравнением четырех сервисов – календарей по выделенным в таблице 1 критериям пользователя-эксперта (то есть меня).

Эксперт выставил свои оценки по основным 5 критериям, таким как: дизайн, удобство использования, функциональность календарного сервиса, динамичность работы, интеграция – по пятибальной шкале (где 1 – это очень плохая оценка, когда продукт не соответствует описанию, 2 – плохое качество и реализация, 3 – удовлетворительная реализация и недостаточный функционал, 4 – неплохой показатель при некоторых недостатках, 5 – идеальное значение показателя).

Ниже получились следующие оценки (так как сервисы достаточно похожие, то и оценки оказались тоже достаточно схожи).

Таблица 2. Расчет средней оценки сервисов-календарей по выделенным критериям

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Критерии/Cервисы | Google - calendar | Teamup | TimeBlocks | Календарь Microsoft Outlook |
| URL-ссылка | <https://calendar.google.com/calendar> | <https://www.teamup.com/> | <https://timeblocks.com/> | <https://www.microsoft.com/en-us/microsoft-365/outlook> |
| Дизайн | 5 | 5 | 4 | 5 |
| Удобство использования | 5 | 4 | 4 | 4 |
| Функциональность календарного сервиса | 5 | 5 | 4 | 5 |
| Динамичность работы | 5 | 5 | 5 | 5 |
| Интеграция | 5 | 4 | 3 | 5 |
| Средняя оценка | 5 | 4,6 | 4 | 4,8 |

Таблица 2. Сравнительная таблица с итоговыми средними баллами оценок сервисов

Итоги и выводы по проведенному сравнительному анализу:

Наивысшую среднюю оценку по итогу сравнения набрал сервис Google Calendar (средняя оценка 5), затем с небольшим отставанием идут сервис календаря Outlook от компании Microsoft (4,8) и свободнораспространяемый календарь TimeUp (оба набрали среднюю оценку 6). Среднюю оценку на уровень хуже получил сервис TimeBlocks (средняя оценка 4,7).

При этом практически все сервисы-календари оказались достаточно удобны, красивы (то есть имеют приятный и утонченный дизайн), информативны и динамичны. Функциональность у них тоже достаточно похожа и ориентирова в основном на пользование специалистами внутри компаний (для планирования в основном рабочих мероприятий).

Так как по итогу сравнения наибольшую оценку набрал сервис календаря от компании Google – то как ориентир для разработки собственного многозадачного календаря я возьму именно его (заостряя внимание на внедрение интеграции не только рабочих, но и учебных, а также личных мероприятий – о чем я говорил выше в разделе «Почему многозадачный календарь действительно полезен для современного человека?»)

Описание набора функций системы, определяющего границы предметной области.

Основные функции разрабатываемой системы:

- зарегистрироваться в системе (пользоваться функционалом только имея учетную запись)

- войти в систему (авторизоваться в приложении с любого другого мобильного устройства)

- создавать группы пользователей, выбирая их тип (рабочая, учебная, личная) – выбирать параметры интеграции пользователей в своей группе

- планировать мероприятия для себя или для группы пользователей, к которой он принадлежит

- рассылать уведомления на устройства приглашенных пользователей

- получать push-напоминания о скором начале ближайших мероприятий

- просматривать свой календарь, удалять из него мероприятия, редактировать их

- добавлять задачи в отдельном разделе приложения, прикреплять их к своим событиям в календаре (либо к другим пользователям)

- получать отчет от календаря за определенный период времени (о том, сколько задач осталось выполнить, сколько времени было проведено в личных/групповых мероприятиях, сколько было отдыха/учебы/работы)

Описание категорий пользователей системы и прецедентов

Система не включает в себя явное разделение пользователей по ролям, то есть все пользователи после регистрации получают статус обычного пользователя, по умолчанию.

Регистрация не требует подтверждения отдельными ответственными лицами (все делается автоматически).

Модерация в календаре тоже никакая не планируется проводиться, поэтому отдельно роль модератора в рамках системы не выделяется.

Но пользователи могут создавать группы (в рамках рабочего или учебных процессов) – в данных группах может быть назначен свой собственный администратор, который имеет возможности назначать задачи другим пользователям, а также планировать мероприятия, выбирая и удаляя с него участников.

Обычно, администратор всегда присутствует в рамках учебной группы (лектор) или рабочей деловой группы (менеджер проекта).

Словарь предметной области (Глоссарий):

Календарь (calendar) – набор событий, задач и мероприятий пользователя в рамках определенного промежутка времени. Графически показывается в виде таблицы, сгруппированной по датам и часовым промежуткам.

Задача (task) – некоторая абстрактная деятельность (или цель - goal), которую пользователь хочет (или должен) выполнить в определенный срок (или в рамках какого-то мероприятия)

Goal (цель) – поставленная задача, отражающая развитие пользователя в рамках деятельности в течение какого-то срока.

Мобильное устройство (mobile device) – смартфон с операционной системой Android, имеющий выход в сеть Интернет и обладающий совместимость к используемому SDK (далее упоминаться будет именно понятие «мобильное устройство»)

Мероприятие (meet) – некоторая совместная деятельность группы пользователей (или одного человека), которое планируется провести в определенное время (оно создается в календаре).

Видеоконференция (или по-простому «созвон» - call) – мероприятие, которое проводится в рамках удаленного аудио или видео - разговора в какой-то программе (например, Google Meet, Zoom, Discord).

One-To-One – созвон двух пользователей в рамках обсуждения локальных задач и достижения целей.

StandUp – созвон множества пользователей из одной деловой (обычно рабочей) группы с целью обсуждения задач на небольшой промежуток времение. Проводится регулярно, может быть циклично запланирован.

Дистанционная лекция – мероприятие учебного типа, проводящееся среди лектора и студентов университета. В отличие от некоторых рабочих мероприятий, в рамках этого события может быть назначен единый администратор для группы.

Функциональная декомпозиция системы

Основные прецеденты:

- Управлять своим аккаунтом (пользователь системы)

- Зарегистрироваться

- Авторизоваться

- Создать группу пользователей

- Выбрать тип группы (учебная, рабочая, личная)

- Добавить участников в группу

- Просматривать визуальный календарь мероприятий

- Просматривать свой календарь (на главной странице)

- Просматривать календарь пользователей в группе

- Создавать задачи

- Распределять задачи в группе по исполнителям

- Прикреплять задачи к определенным мероприятиям

- Запланировать мероприятие

- Выбрать тип мероприятия

- Выбрать время и продолжительность

- Добавить пользователей из группы на мероприятие

- Отослать уведомления о приглашении на мероприятие

- Получать уведомления о мероприятиях

- Получать уведомление о приглашении на мероприятие

- Получать дополнительно напоминание о наступлении мероприятия (за 10 мин до начала, например)

- Получать отчеты своем календаре (опционально, по событиям или задачам)

Диаграмма прецедентов в нотации UML изображена ниже на рисунке 2:

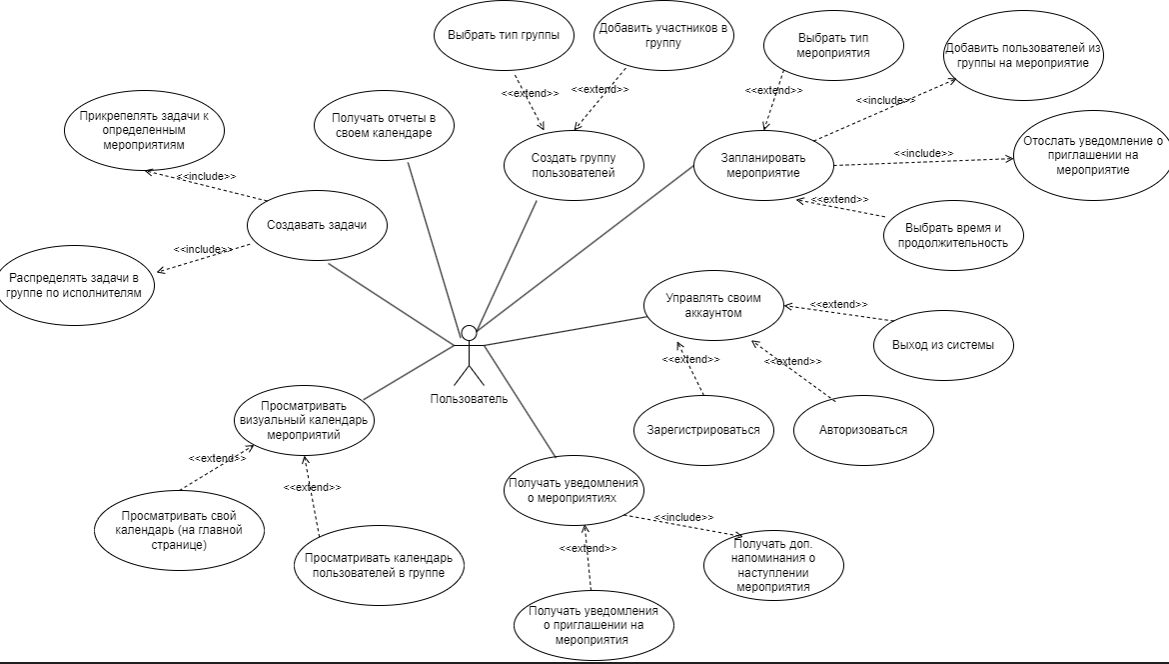


Рисунок 2. Диаграмма прецедентов

Описание некоторых прецедентов

Прецедент «Зарегистрироваться»

Предусловие: Пользователь решил впервые воспользоваться мобильным приложением «Многофункциональный календарь»

Действующее лицо: пользователь

Основной поток: пользователь открывает приложение, нажимает кнопку «Зарегистрироваться», вводит учетные данные для регистрации, затем получает сообщение по указанному номеру с кодом, который подтверждает в приложении.

Альтернативный поток:

Пользователь ввел неверный код подтверждения из СМС. Тогда он может пройти основной поток еще раз в течении 2 дополнительных попыток.

Пользователь уже зарегистрирован в системе. Тогда при попытке зарегистрироваться с уже сохраненными в системе учетными данными, он будет уведомлен о наличии зарегистрированной учетной записи с введенными данными.

Постусловие: При наличии зарегистрированной учетной записи, пользователь будет перенаправлен на страницу авторизации.

Прецедент «Авторизоваться»

Предусловие: Пользователь решил воспользоваться функционалом системы при уже имеющейся учетной записи, то есть пользователь уже зарегистрирован в системе.

Действующее лицо: пользователь

Основной поток: пользователь открывает мобильное приложение, на главной странице видит раздел для авторизации, затем вводит свои данные и получает доступ к своей личной учетной записи и к личному кабинету.

Альтернативный поток: Пользователь ещё не зарегистрирован в системе. Тогда при попытке авторизоваться с некоторыми данными, он будет уведомлен об отсутствии требуемой учетной записи.

Постусловие: При отсутствии зарегистрированной учетной записи, пользователь будет перенаправлен на страницу регистрации.

Описание состава данных

Выделим следующие абстракции данных (сущности системы):

- User (пользователь с некоторыми учетными данными)

- Group (часть некоторого сообщества, к которой прикреплен пользователь)

- Event (событие (или мероприятие), предполагающее встречу/созвон – или какую-то другую совместную деятельность пользователей из одной группы)

- Task (задача, поставленная на исполнение администратором группы другому пользователю (или себе))

- Report (отчет о мероприятиях, событиях, задачах в календаре пользователя за определенный промежуток времени).

- Calendar (составная сущность – представляет собой отображение (или правильнее сказать группировку) события и задач пользователя по промежуткам времени). Планируется создать отдельную активность для отображения своего календаря (или календаря любого другого пользователя из той же группы). На диаграмме классов не отображается (так как создается средствами ORM на уровне SQL базы данных – для связи «многие ко многим»).

Дополнительно используются перечисления GroupType (тип группы: рабочая, учебная, личная), TaskType (какой – то абстрактный тип задачи: например, постановка цели (goal), присутствие на встречах, выполнение задания), EventType (тип события: Personal (Solo), OneToOne, StandUp, Meet и так далее).

Диаграмма классов (уровня проектирования) системы в нотации UML представлена ниже на рисунке 3:

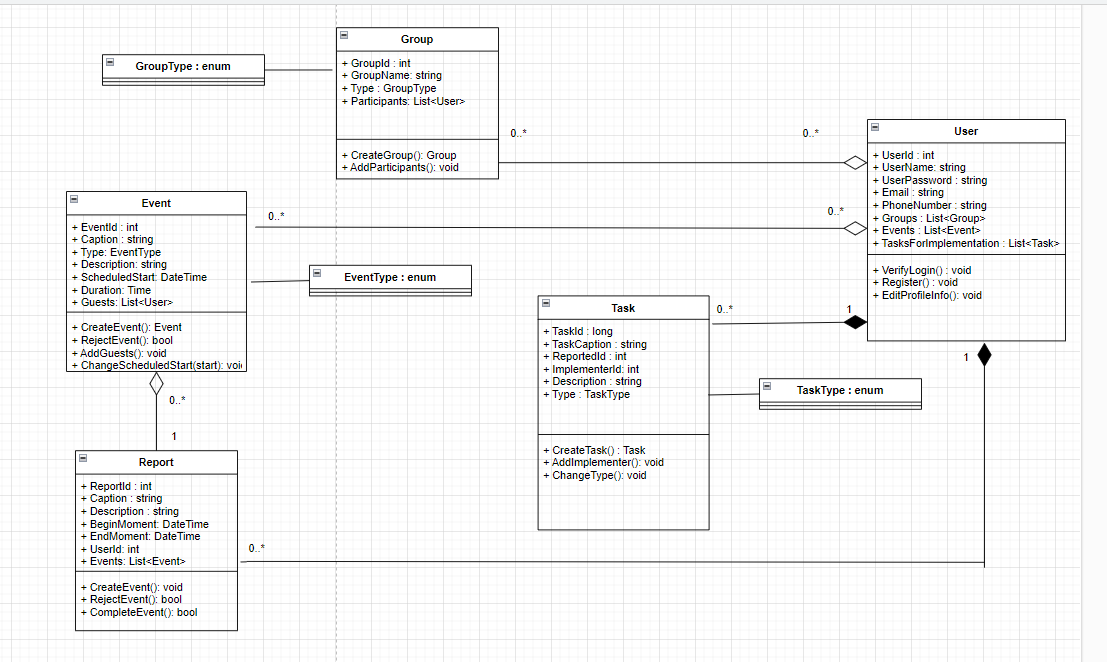


Рисунок 3. Диаграмма классов уровня проектирования

Диаграмма схемы базы данных в нотации ER представлена ниже на рисунке 4:

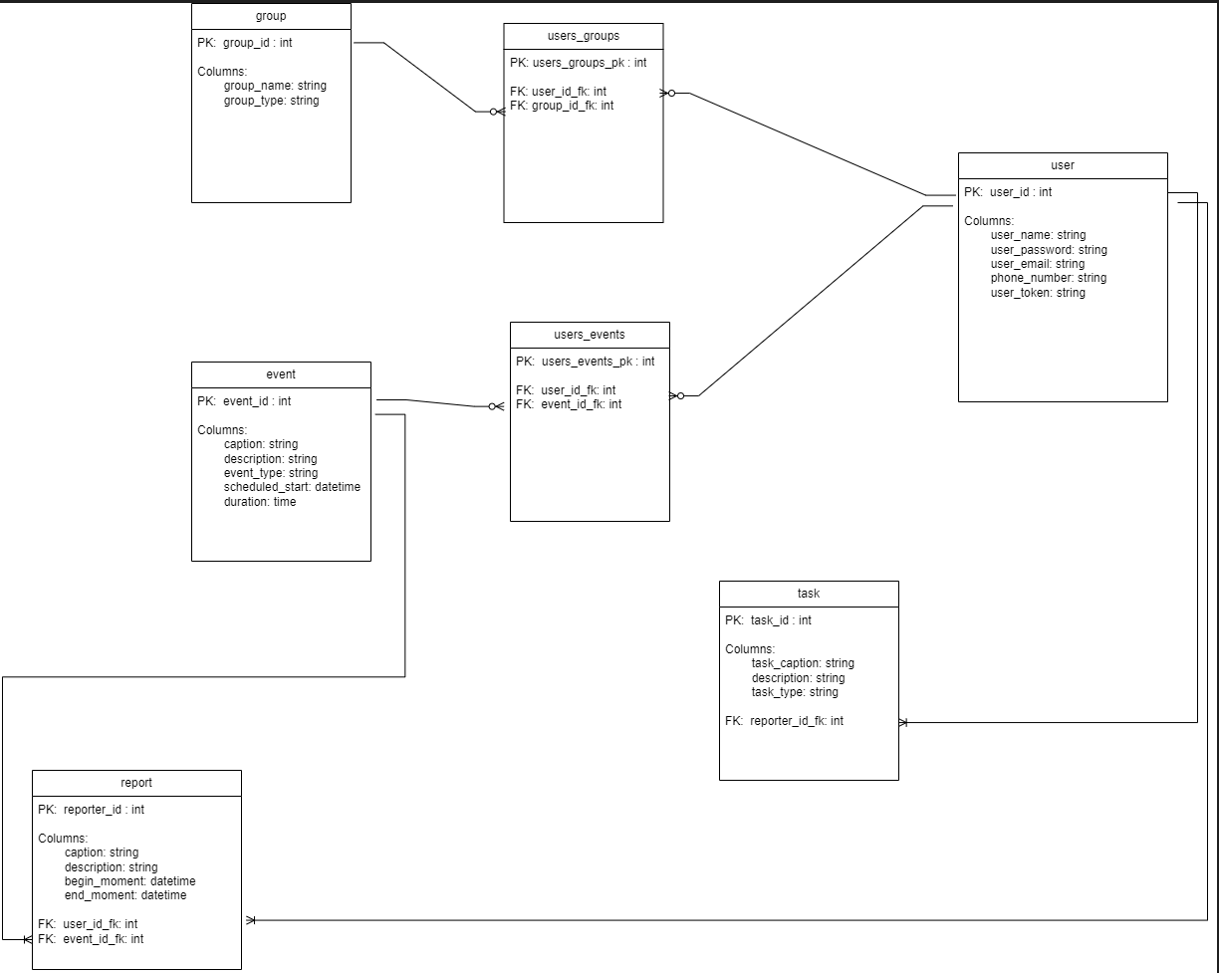


Рисунок 4. Схема базы данных

## Требования к разрабатываемой системе

Выделим основные категории требований, которые необходимо предъявить к анализируемой выше разрабатываемой системе:

1. Функциональные требования

Список функциональных требований, минимально необходимых для прототипа мобильного приложения:

1.1. Функция 1

1.1.1 Название - возможность использовать авторизацию пользователей с серверной аутентификацией

1.1.2 Описание - пользователь получает доступ к функционалу на сервере с помощью специального сгенерированного токена, получаемого после авторизации (после входа в систему)

1.1.3 Назначение – предозначить доступ только запросам от авторизованных пользователей.

1.1.4 Зависимости – фреймворк ASP.NET Core 5, Google JWT Authentification

1.2. Функция 2.

1.2.1 Название - регистрации пользователя

1.2.2 Описание – возможность регистрации пользователя в системе

1.2.3 Назначение – возможность сохранять учетную запись пользователя

1.3. Функция 3.

1.3.1 Название – подтверждение учетной записи

1.3.2 Описание – возможность подтверждения регистрации новой учетной записи пользователя путем ввода кода, высылаемого на электронную почту

1.3.3 Назначение – верифицировать используемую в настройках учетной записи электронной почты

1.4. Функция 4.

1.4.1 Название – создание пользовательских групп

1.4.2 Описание – возможность создания пользователем групп различных типов (рабочих, учебных, личных), добавления в него пользователей.

1.4.3 Назначение – сохранение групп с пользователями для создания общих мероприятий и назначения задач на исполнение участникам группы

1.5. Функция 5.

1.5.1 Название – создание пользовательских мероприятий

1.5.2 Описание - возможность создавать мероприятия, приглашать туда других пользователей (мероприятия будут проводиться во внешних системах проведения видеоконференций – например, Google Meet или Discord)

1.5.3 Назначение – сохранение групповых мероприятий с ссылками на проведение их в системах видеоконференции

1.6. Функция 6.

1.6.1 Название - создавать пользовательские задачи

1.6.2 Описание – возможность прикреплять к пользователям задачи, синхронизировать их с событиями в их календарях

1.6.3 Назначение – сохранение пользовательских задач в базе данных

1.7. Функция 7

1.7.1 Название – создавать пользовательские отчеты

1.7.2 Описание – возможность создавать отчеты по задачам/мероприятиям (для анализа занятости пользователя)

1.7.3 Назначение – просмотр и сохранение пользовательских отчетов в базе данных

1.8. Функция 8.

1.8.1 Название – получать уведомления о мероприятиях

1.8.2 Описание – возможность организации системы уведомлений и напоминаний (о приглашении на групповое мероприятие, приближение начала данного мероприятия) для каждого пользователя-участника мероприятия

1.8.3 Назначение – получение mail-сообщений на свой адрес электронной почты

1.8.4 Зависимость – библиотека System.NET.Mail.Kit и средства работы с SMTP - сервером

2. Нефункциональные требования (то есть требования, определяющие свойства, которые система должна демонстрировать, или ограничения, которые она должна соблюдать, не относящиеся к поведению системы):

2.1. Производительность - трафик запросов (не менее 10000 одновременных запросов от разных пользователей в небольшой промежуток времени)

2.2. Надёжность (система работает 24 на 7 и предоставляет возможность для работы с ней)

2.3. Платформо-ориентированность (возможность запускать приложение на операционных системах семейства Android)

2.4. Обратная совместимость (поддержка всех устройств Android с системой не ниже 11 версии – то есть API 31 и больше)

2.5. Версионирование - использование Android SDK версии не ниже версии Android 11

2.6. Масштабируемость системы.

ПРОЕКТИРОВАНИЕ СИСТЕМЫ

Выбранный вариант разработки программной системы

- Использование обозначенных библиотек и фреймворков

- Разработка серверной части с использованием ASP.NET Core на платформе .NET 7. Для работы с БД – Entity Framework.

- Разработка клиентской части – Flutter и язык программирования Dart.

- Использование обозначенных прикладных средств разработки:

- IDE MS Visual Studio (разработка серверной части)

- Postman (тестирование HTTP-запросов сервера)

- IDE Android Studio (разработка клиентской части и верстка приложения)

Ниже на рисунке 5 представлена диаграмма развертывания описанной выше программной системы:

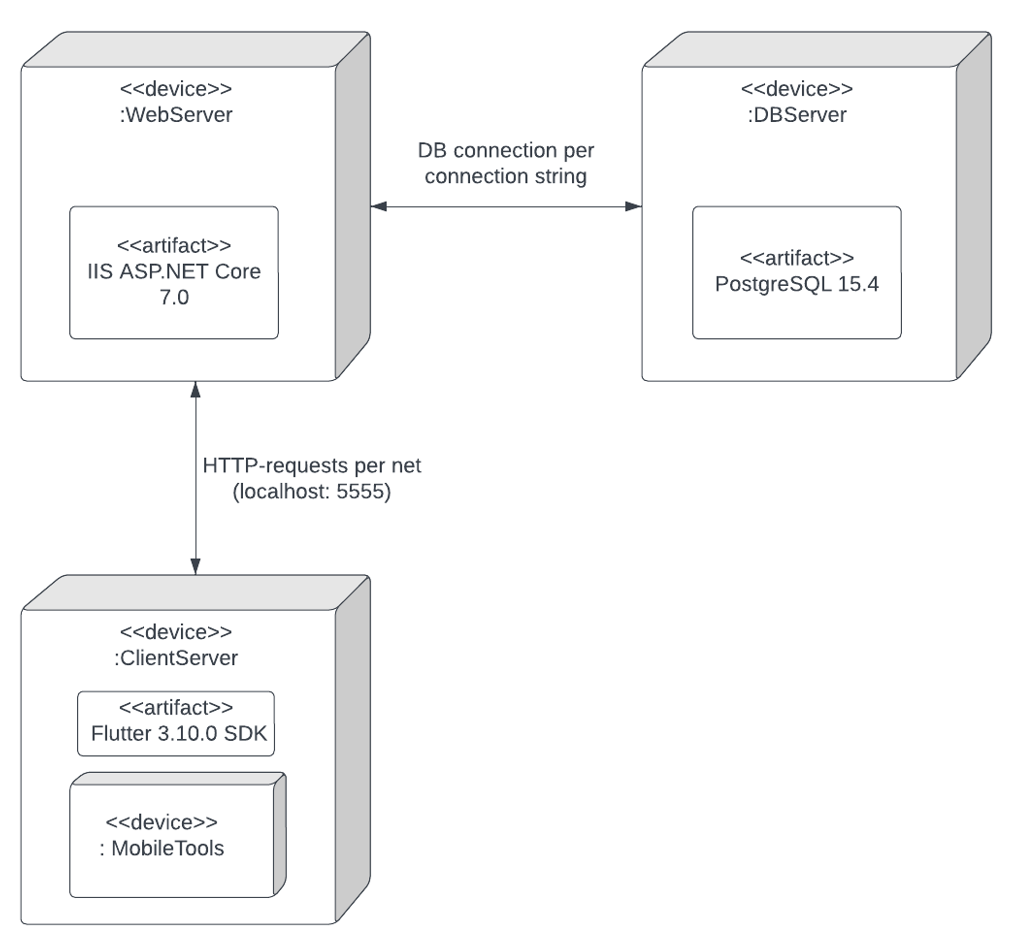


Рисунок 5. Диаграмма развертывания ПС

Построение схемы работы алгоритма (алгоритм регистрации пользователя в системе представлен ниже на рисунках 6.1 – 6.2):

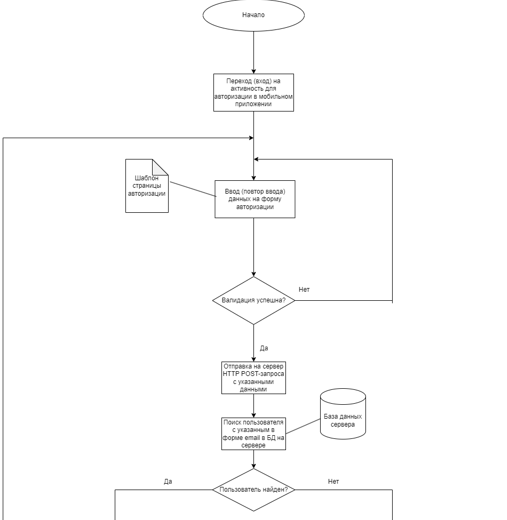


Рисунок 6.1. Схема алгоритма авторизации пользователя (часть 1)

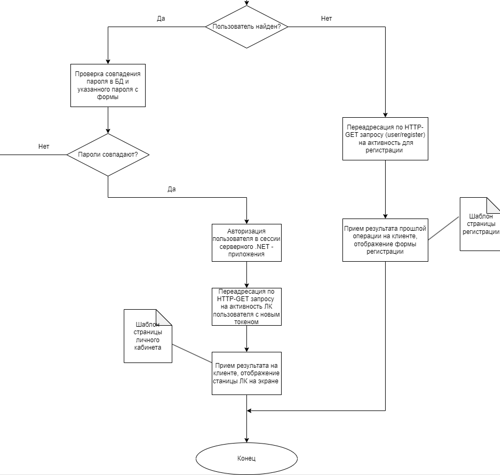


Рисунок 6.2. Схема алгоритма авторизации пользователя (часть 2)

РАЗРАБОТКА ПРИЛОЖЕНИЯ

План проведения разработки ПС мобильного приложения календаря:

- Разработка сервиса приложения (ASP.Net Core 7.0 Framework)

Github: <https://github.com/Tigeroff2002/ToDoCalendarServer>

- Разработка клиентской части приложения (Flutter 3.10.0 Framework)

Github: <https://github.com/Tigeroff2002/ToDoCalendar>

# ТЕСТИРОВАНИЕ СИСТЕМЫ

Тестирование запросов к серверному приложению

Для тестирования HTTP-запросов, поступаемых «внутрь и наружу» серверного приложения я использовал программу Postman.

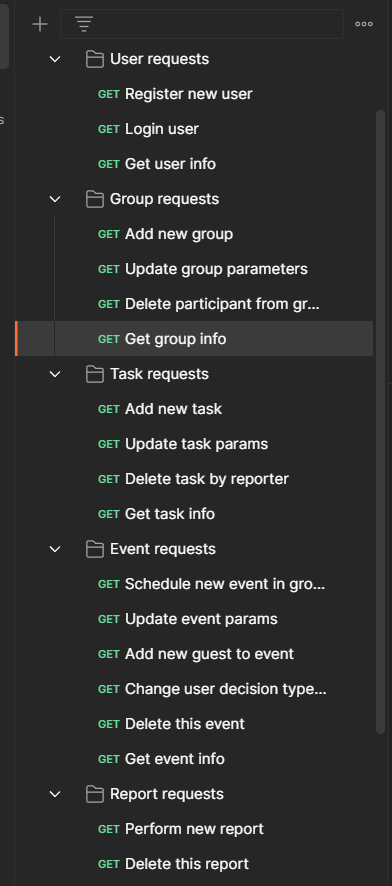


Рисунок 7. Список коллекций запросов

Запросы к endpointу /users:

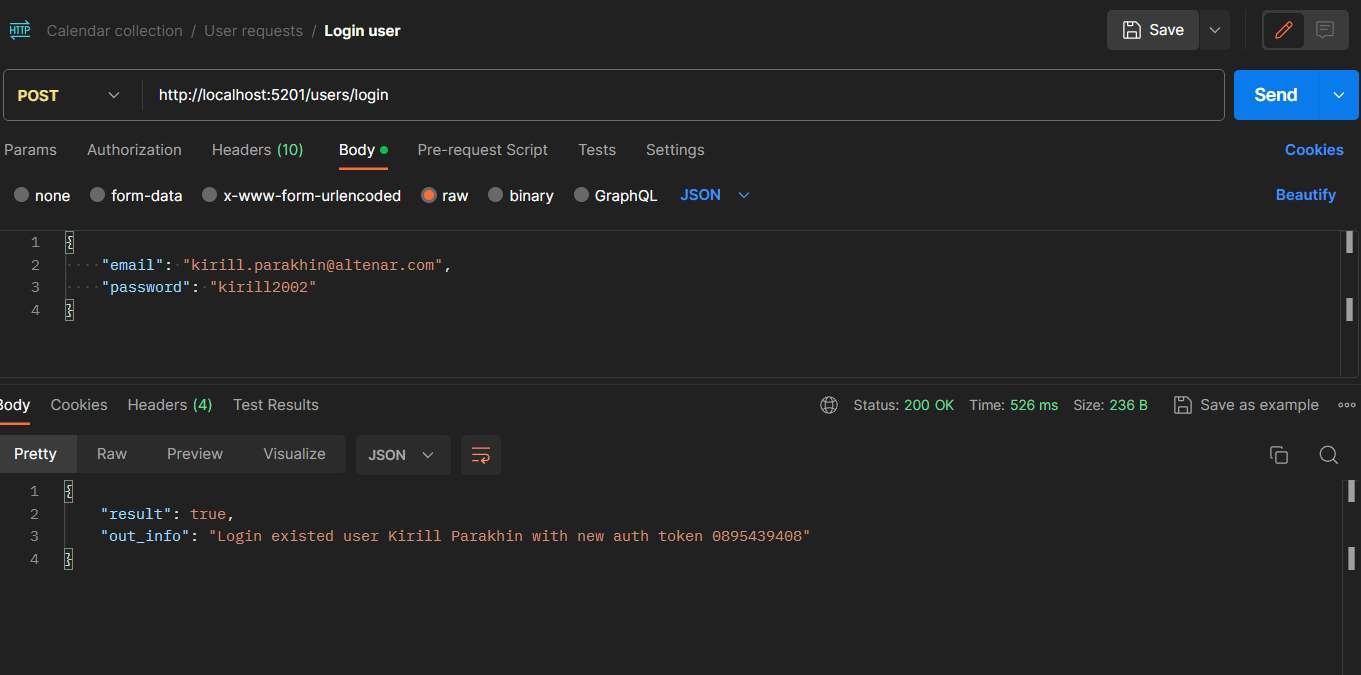


Рисунок 8.1. Тестирование запроса на авторизацию пользователя

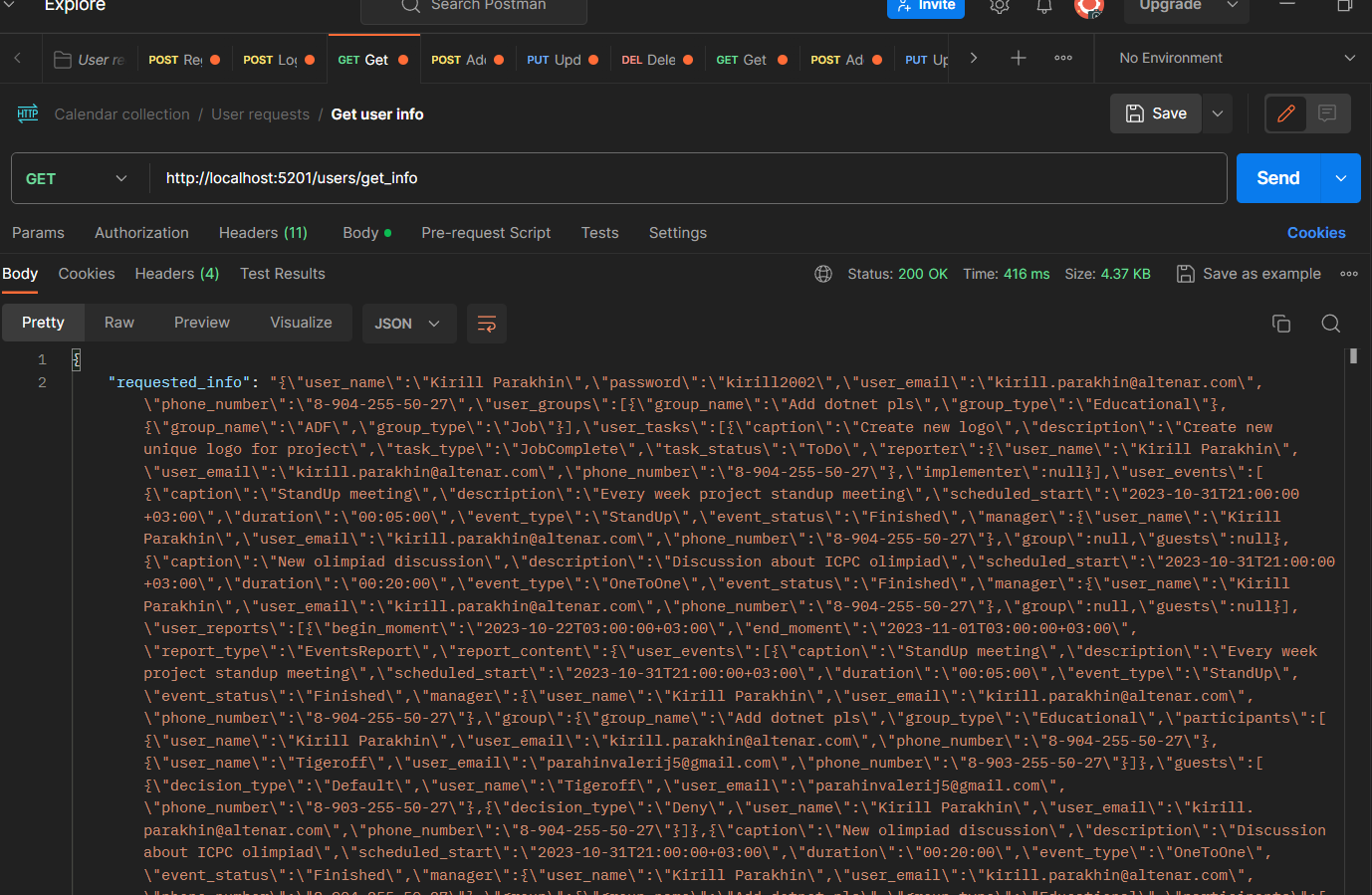


Рисунок 8.2. Тестирование запроса на получение полной информации о пользователе

Запросы к endpointу /groups:

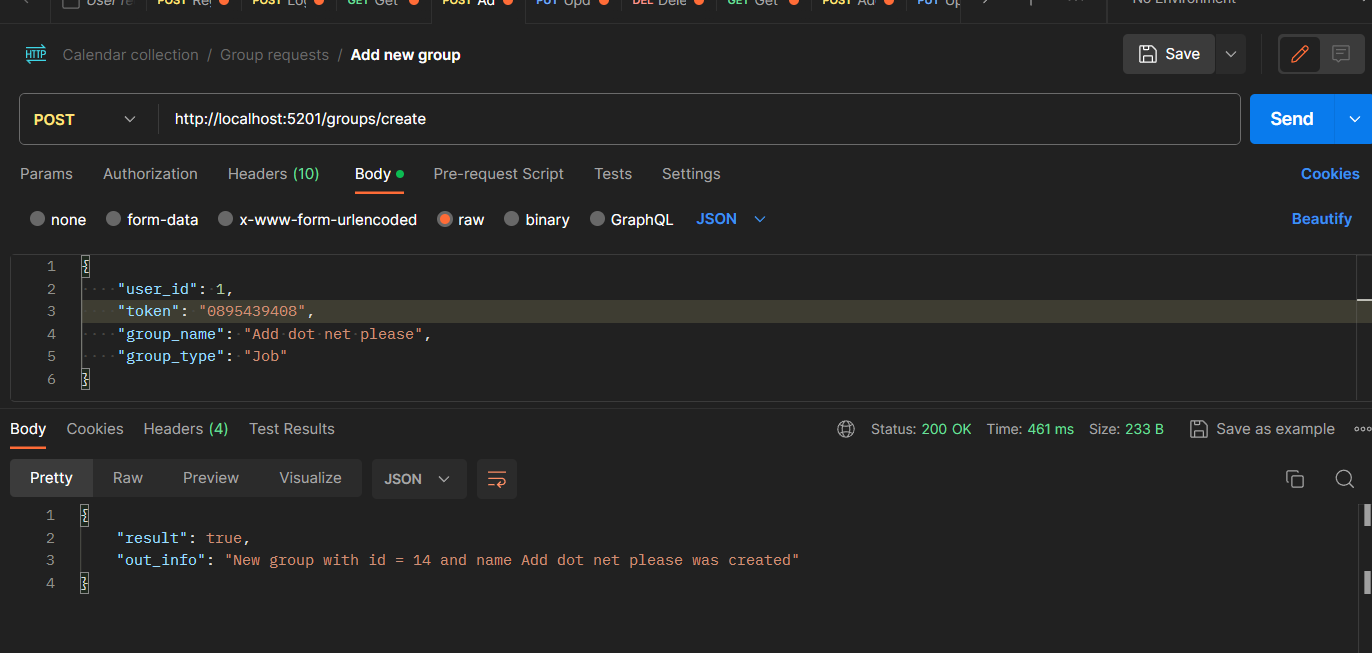


Рисунок 8.3. Тестирование запроса по созданию новой группы

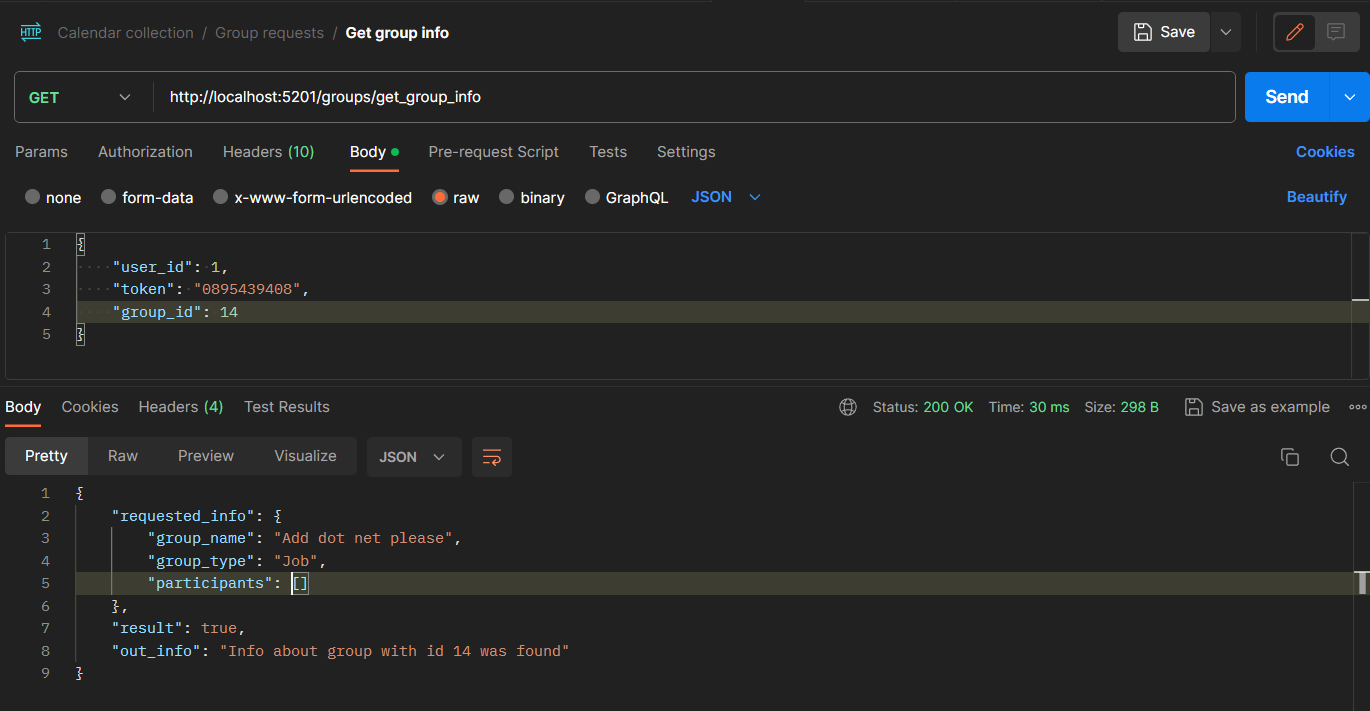


Рисунок 8.4. Тестирование запроса получения информации о добавленной группе

Запросы к endpointу /events:

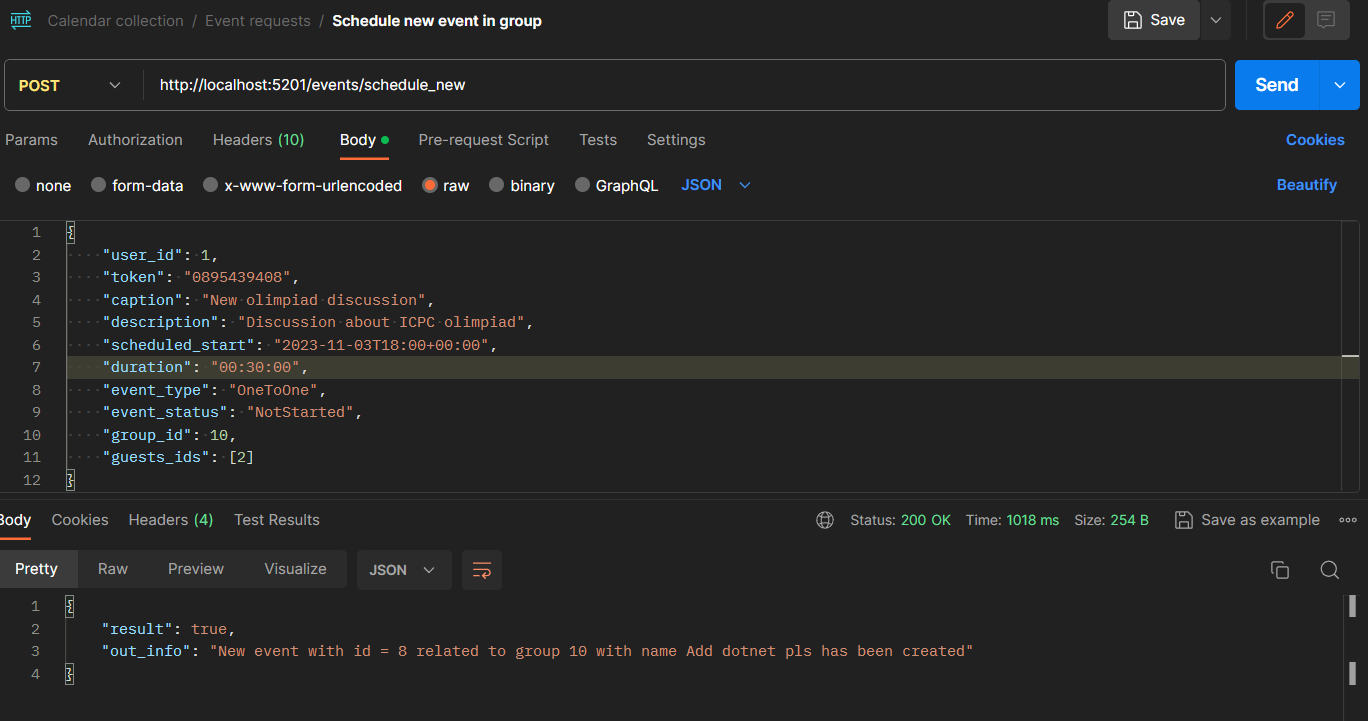


Рисунок 9.1. Тестирование запроса на создание нового мероприятия

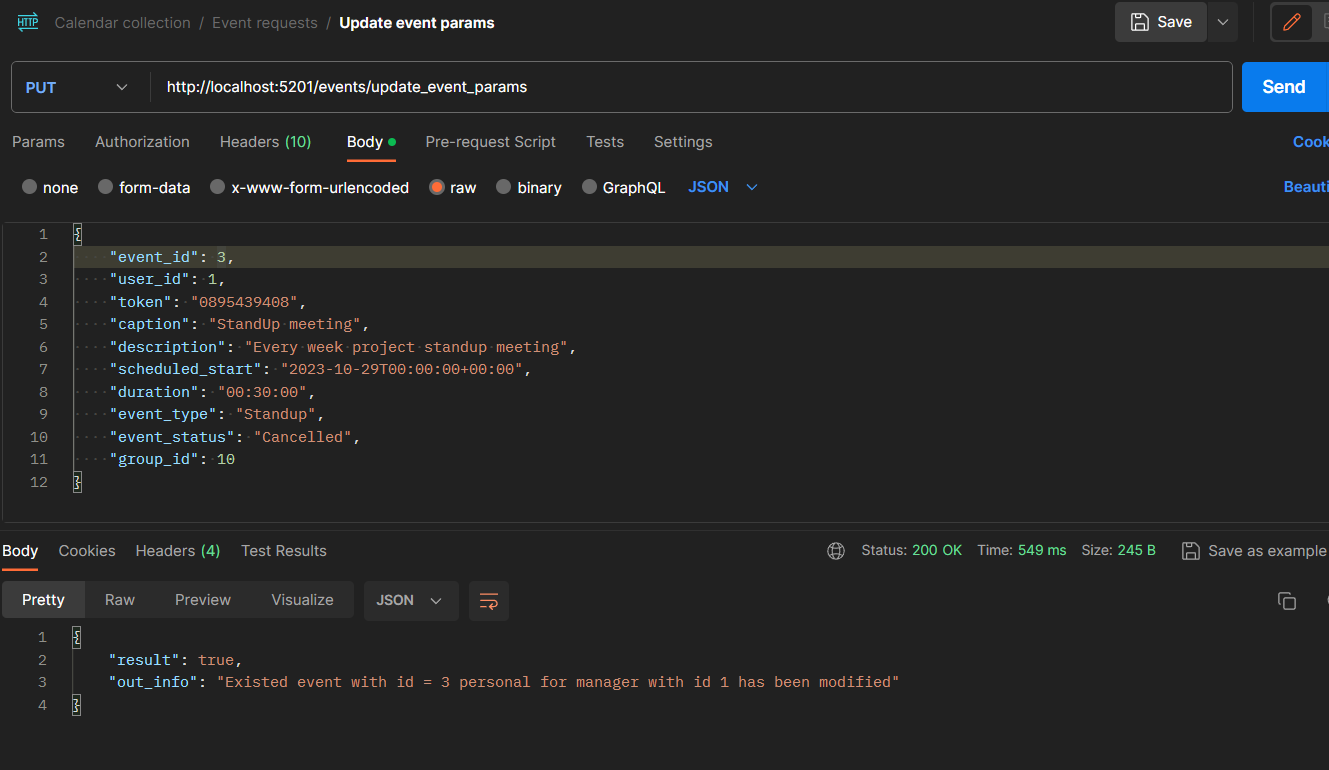


Рисунок 9.2. Тестирование запроса на обновление параметров мероприятия

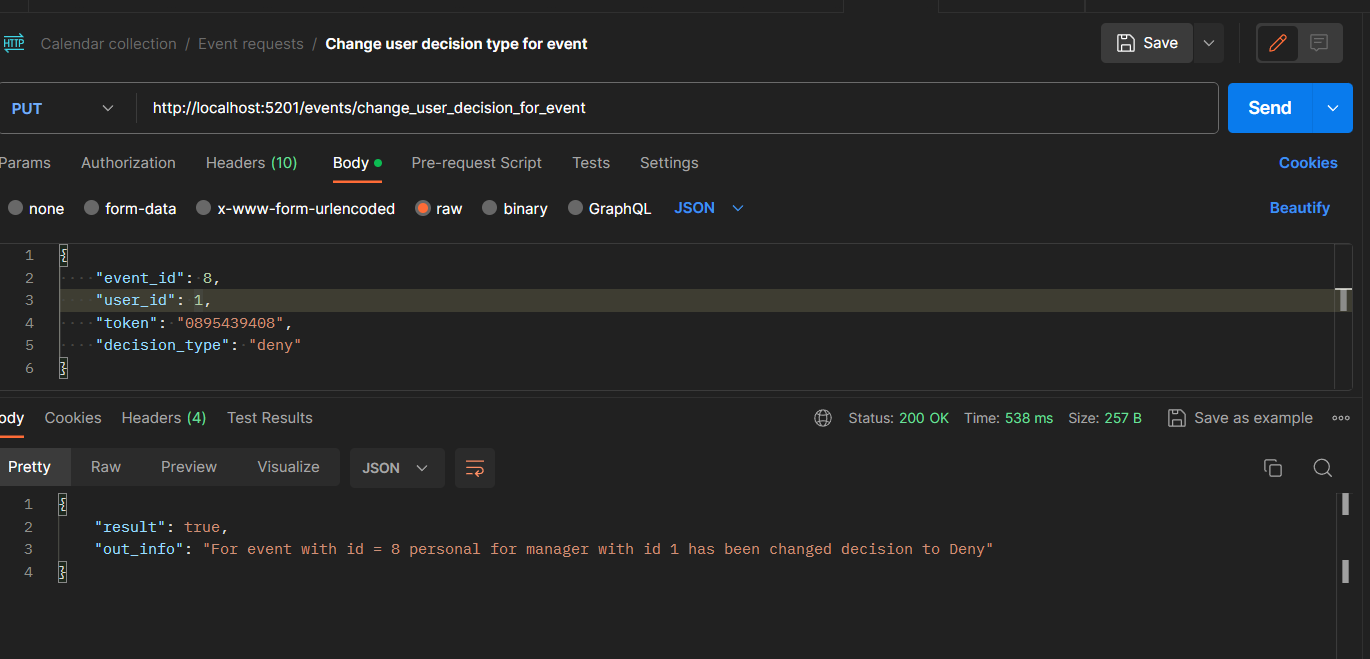


Рисунок 9.3. Тестирование запроса на изменение решения о посещении мероприятия пользователем

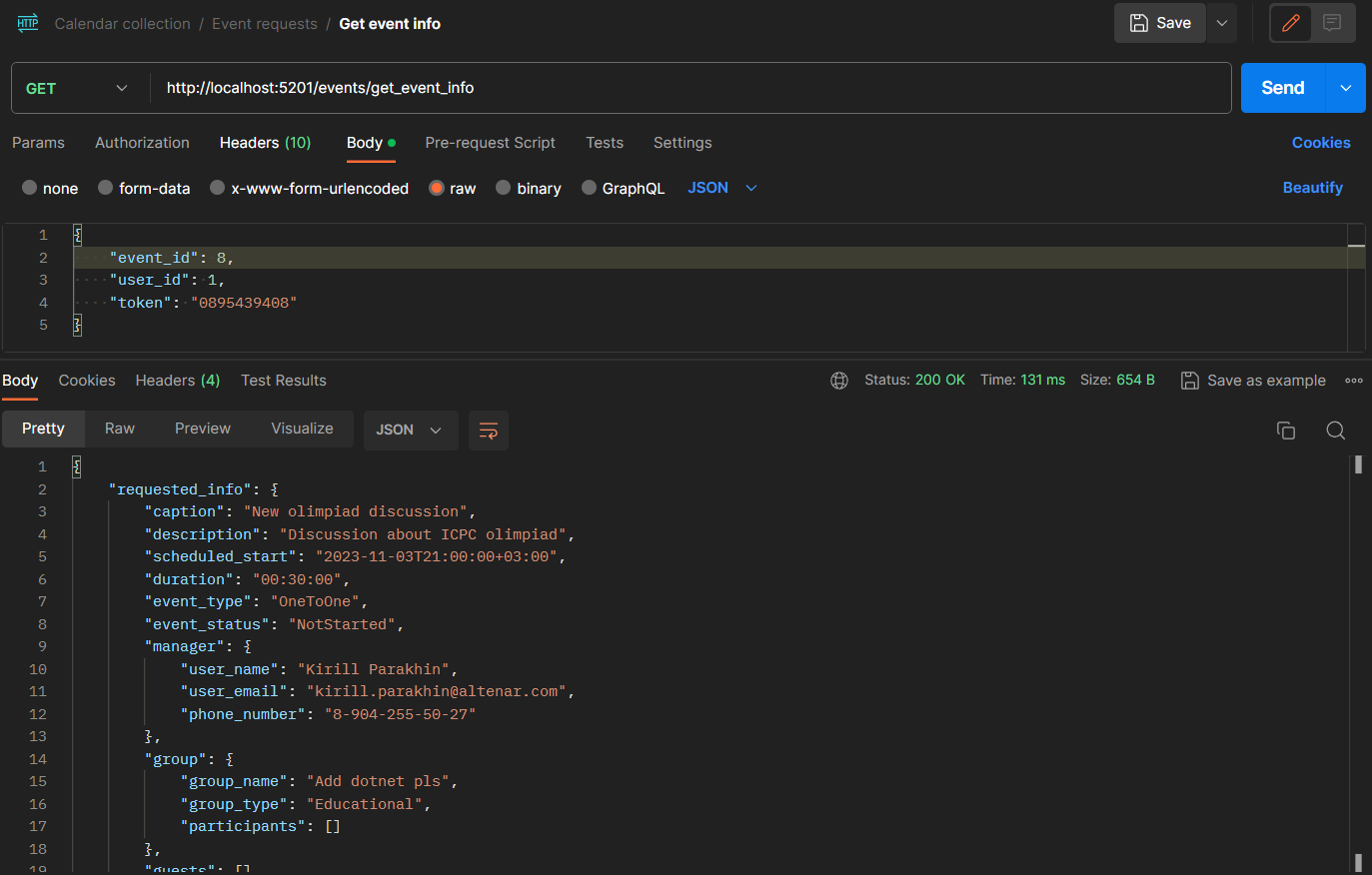


Рисунок 9.4. Тестирование запроса на получение полной информации о предприятии

Запросы к endpointу /tasks:

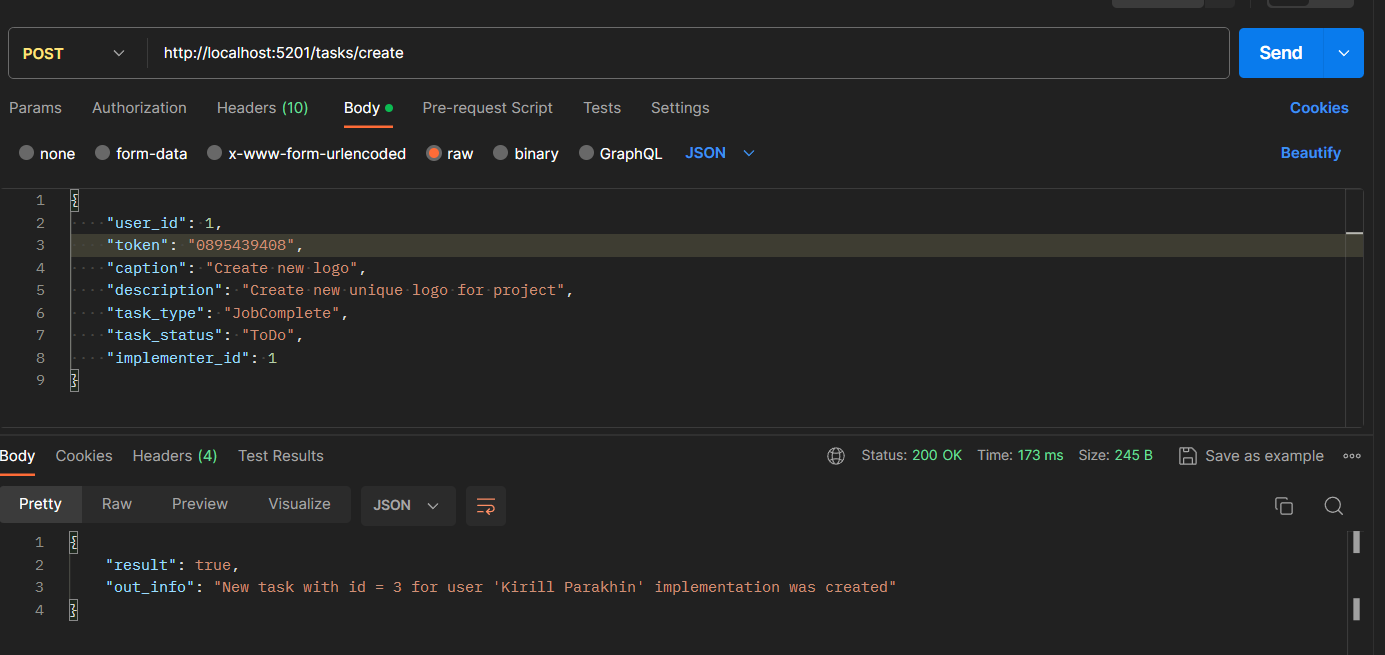


Рисунок 10.1. Тестирование запроса на создание новой задачи

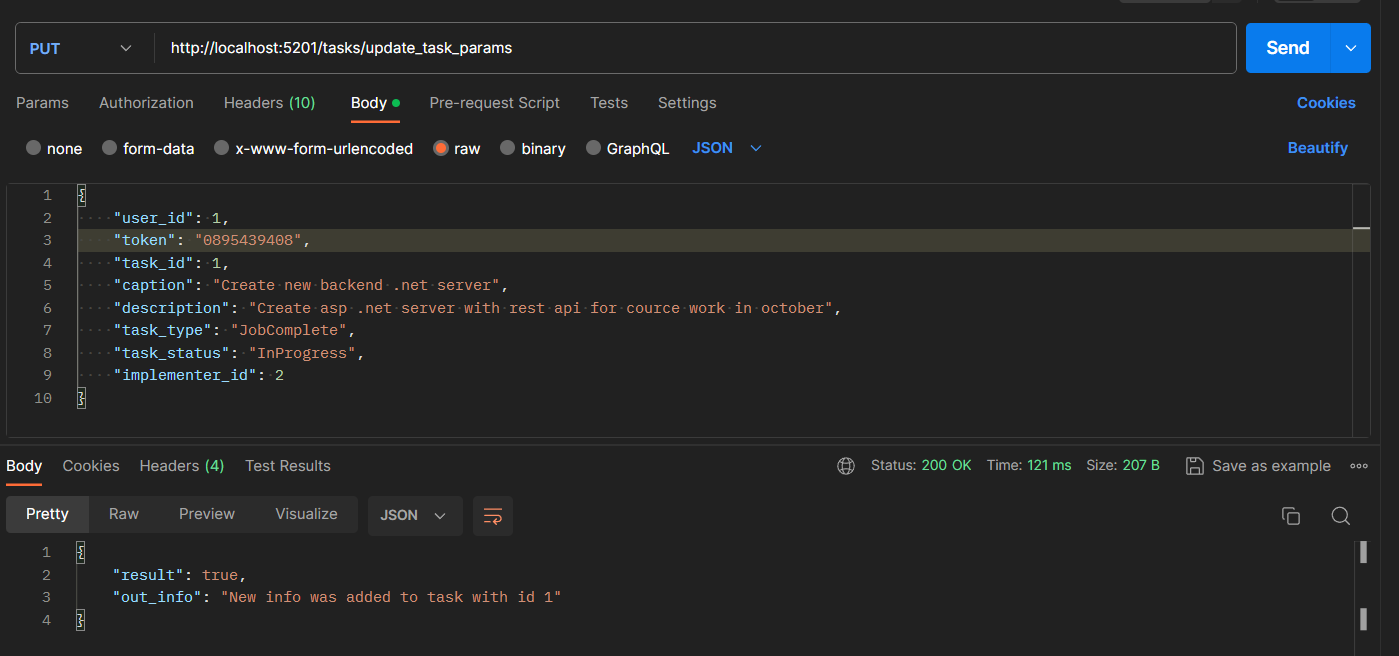


Рисунок 10.2. Тестирование запроса на обновление параметров задачи

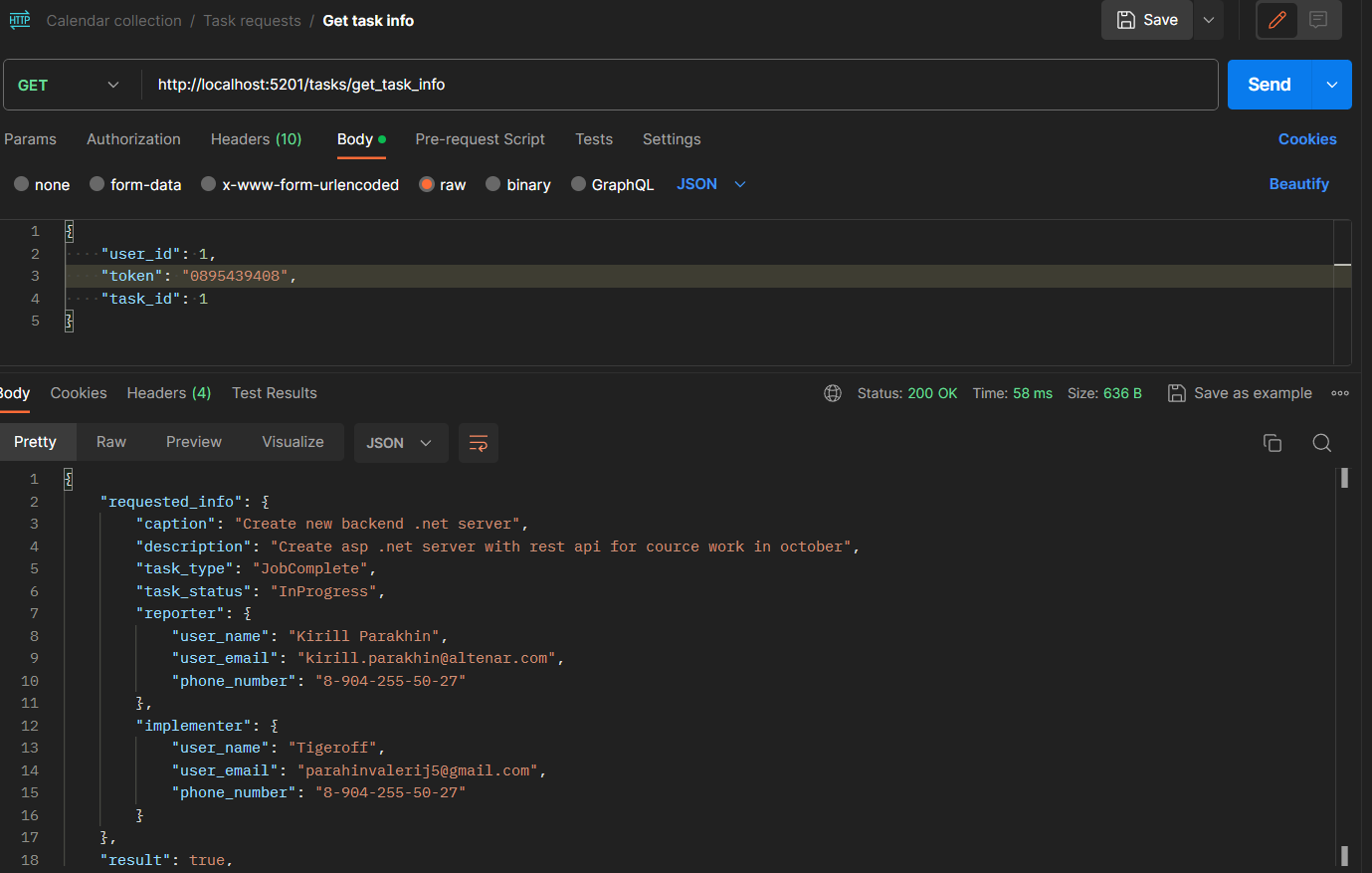


Рисунок 10.3. Тестирование запроса на получение информации о текущей задаче

Запросы к endpointу /reports:

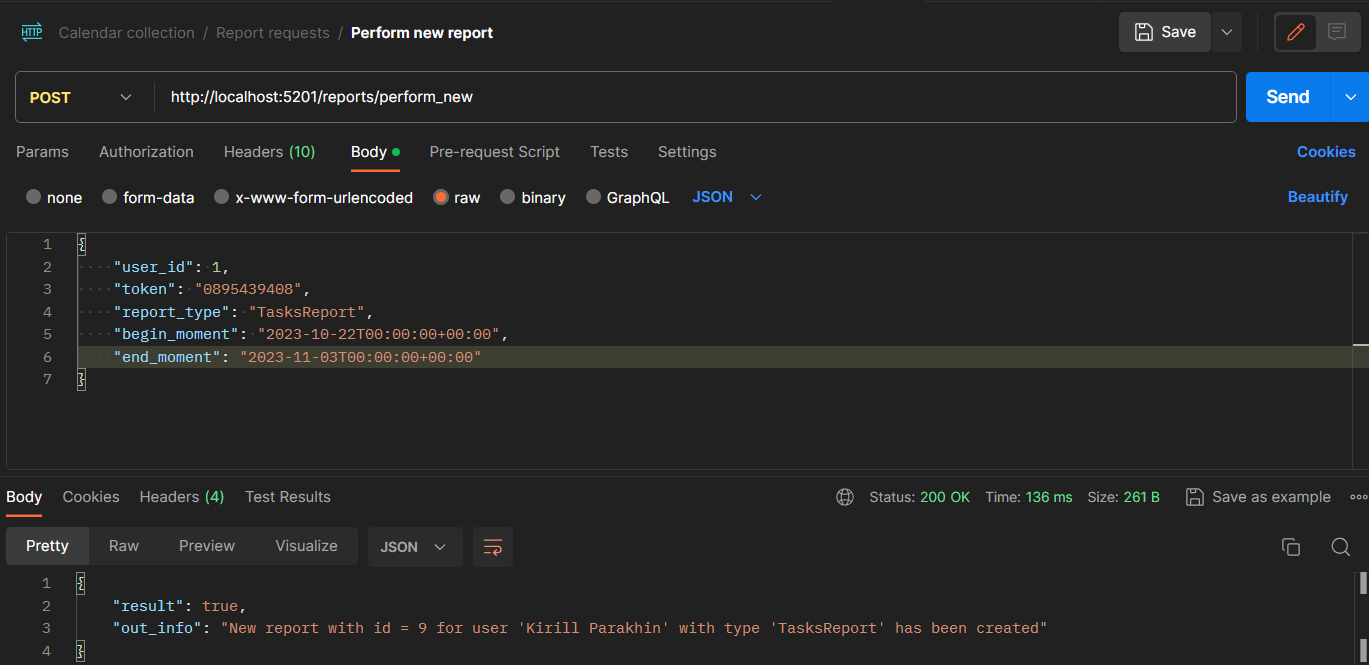


Рисунок 11.1. Тестирование запроса на создание нового отчета (по задачам)

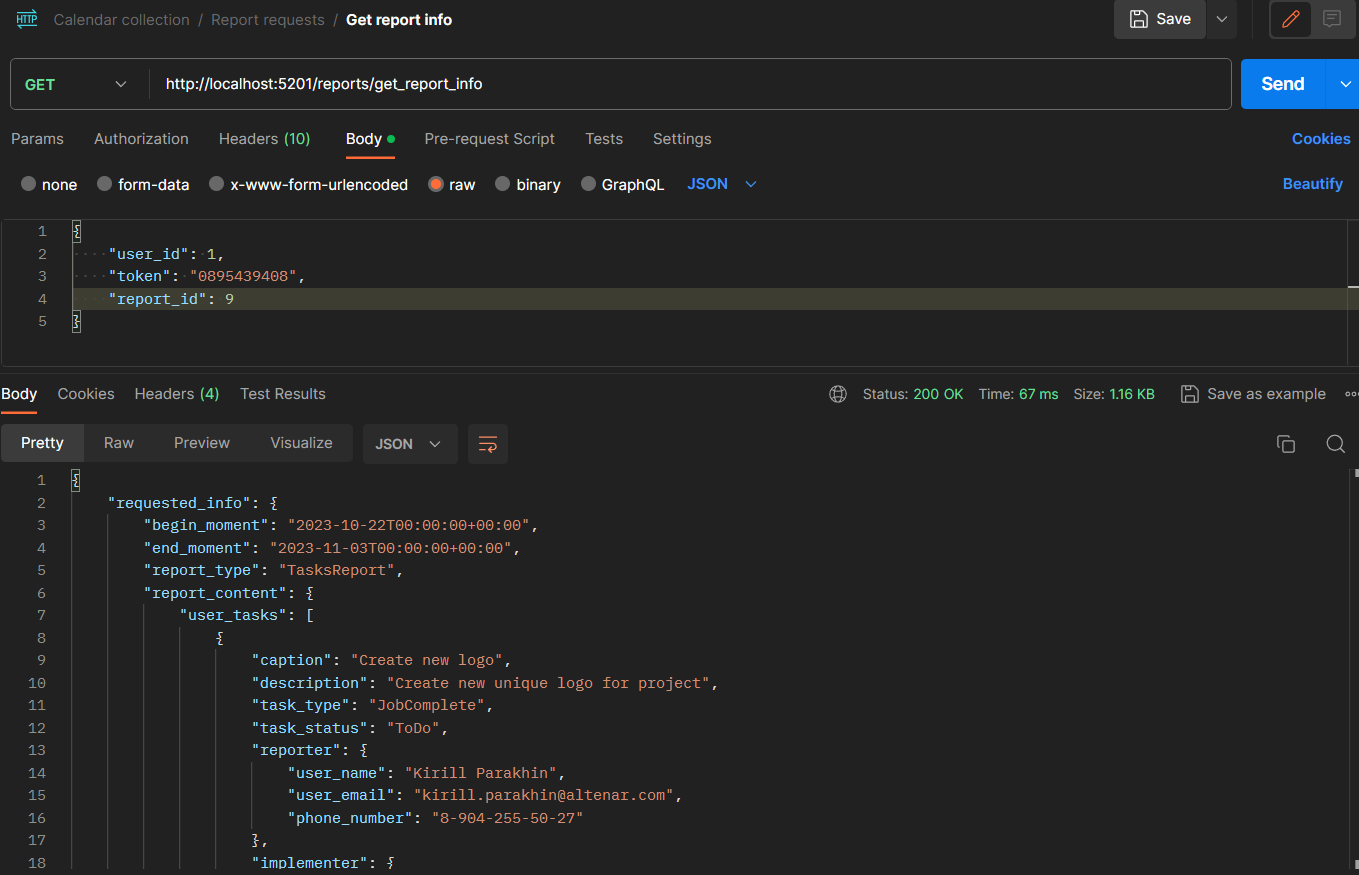


Рисунок 11.2. Тестирование запроса на получение информации из созданного отчета по задачам

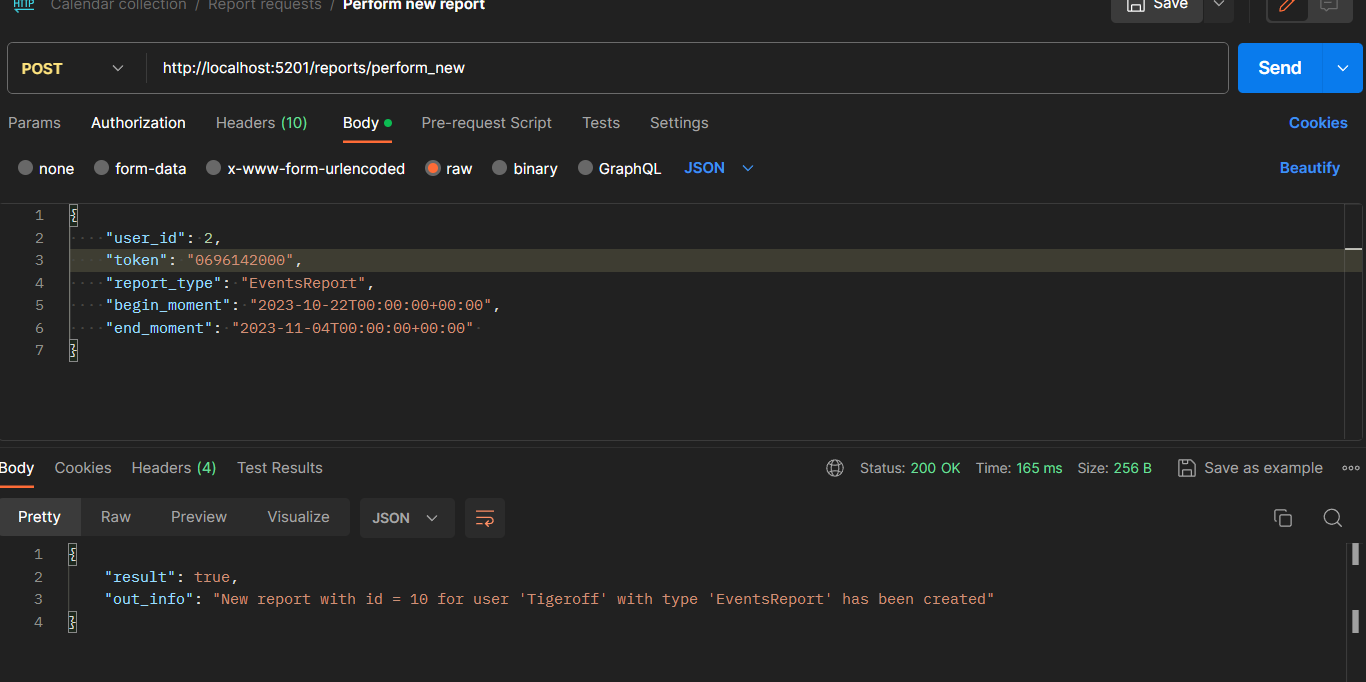


Рисунок 11.3. Тестирование запроса на создание нового отчета (по мероприятиям)

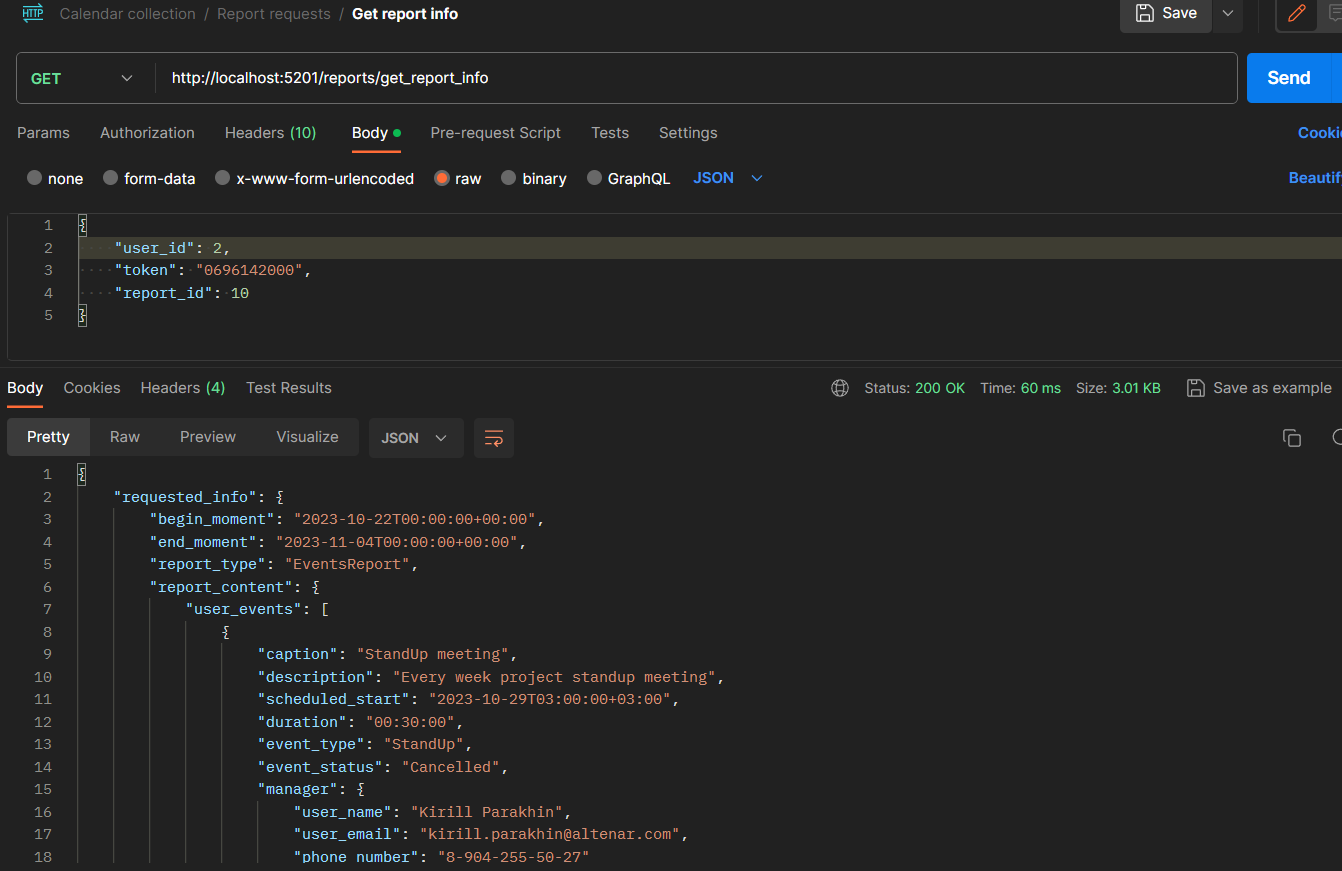


Рисунок 11.4. Тестирование запроса на получение информации из созданного отчета по мероприятиям

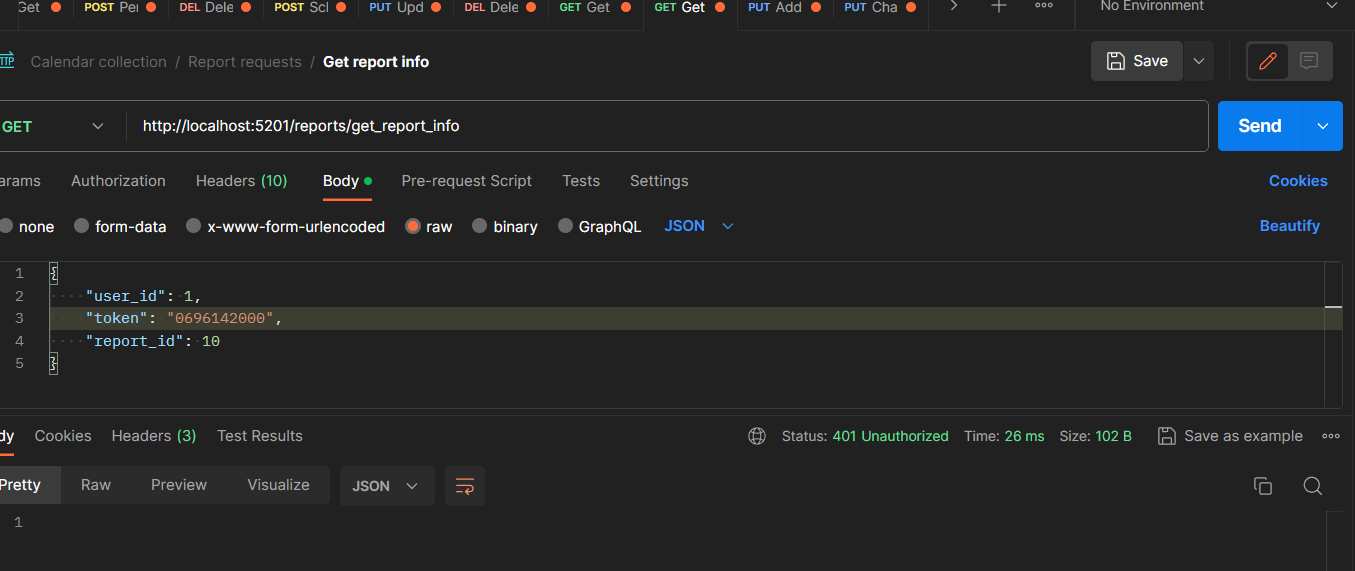


Рисунок 11.5. Тестирование запроса пользователя с несоответствующим токеном (ответ 401 Unathorized)

Разработка клиентского приложения на фреймворке Flutter

## Модульное тестирование

Для написания unit-тестов для серверного приложения на платформе .NET я использовал основные фреймворки для проведения модульного тестирования (xUnit, Moq, FluentAssertions).

Разработка логотипа приложения

Для прототипа мобильного приложения «Многозадачный календарь» при помощи сервиса FreeLogoDesign был разработан собственный логотип, который включает индивидуальный ник разработчика (Tigeroff) – рисунок 7.



Рисунок 7. Разработка логотипа для мобильного приложения Tigeroff Calendar

## 

## Репозиторий проекта

Вся разработка велась с использованием системы контроли версий git и сервиса для хранения удаленных репозиториев Github, на котором размещен код моей программной системы по ссылке: <https://github.com/Tigeroff2002/ToDoCalendar> (рис.6 – представлен скришот курсового проекта, расположенный на сайте GitHub).

Рисунок 17. Репозиторий проекта на GitHub

# Заключение

В ходе выполнения курсового проекта была разработано мобильное приложение, предоставляющее функционал удобного многозадачного календаря пользователя.

# Список использованных источников

1. Официальная документация Android Studio. [Электронный ресурс] URL: https://developer.android.com/guide

2. Официальная документация React Native. [Электронный ресурс] URL: <https://reactnative.dev/docs/getting-started>

3. Гриффитс, Д. Head First. Программирование для Android /Д. Гриффитс. – СПб.: Питер, 2016. – 704 с.

4. Нативная разработка мобильных приложений [Электронный ресурс] / Льюис Ш., Данн М., пер. с англ. А.Н. Киселева. - М. : ДМК Пресс, 2020.

5. Разработка мобильных приложений в среде Android Studio : учебное пособие [Электронный ресурс] / Л. В. Пирская. – Ростов н/Д : ЮФУ, 2019

ПРИЛОЖЕНИЕ А. Схема работы алгоритма